```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
data = pd.read_csv("/content/telecom_customer_churn.csv")

# Drop irrelevant columns like Customer ID, City, Zip Code, Latitude,
Longitude
data.drop(columns=['Customer ID', 'City', 'Zip Code', 'Latitude',
'Longitude'], inplace=True)

# Handling missing values (if any)
data.fillna(method='ffill', inplace=True)

# Encode categorical features
label_encoders = {}
for column in data.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    data[column] = label_encoders[column].fit_transform(data[column])

# Split the dataset into features and target variable
X = data.drop(columns=['Churn Category', 'Churn Reason'])
y = data['Churn Category']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train the model (Random Forest Classifier)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_rf = rf_model.predict(X_test)

# Evaluate the model
```

```
print("Random Forest Classifier:")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("Classification Report:")
print(classification_report(y_test, y_pred_rf))

# Train the model (XGBoost Classifier)
xgb_model = XGBClassifier(random_state=42)
xgb_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb = xgb_model.predict(X_test)

# Evaluate the model
print("\nXGBoost Classifier:")
print("Accuracy:", accuracy_score(y_test, y_pred_xgb))
print("Classification Report:")
print(classification_report(y_test, y_pred_xgb))
```

```
OUTPUT:
Random Forest Classifier:
Accuracy: 0.42796309439318664
Classification Report:
              precision    recall  f1-score   support

           0       0.03      0.00      0.01       210
           1       0.44      0.96      0.60       624
           2       0.24      0.02      0.03       242
           3       0.00      0.00      0.00       149
           4       0.00      0.00      0.00       184

    accuracy                           0.43      1409
   macro avg       0.14      0.20      0.13      1409
weighted avg       0.24      0.43      0.27      1409

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py
:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py
:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
```

```
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py
:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
XGBoost Classifier:
Accuracy: 0.3924769339957417
Classification Report:
              precision    recall  f1-score   support

           0       0.21      0.09      0.13       210
           1       0.44      0.81      0.57       624
           2       0.21      0.09      0.13       242
           3       0.09      0.01      0.02       149
           4       0.12      0.03      0.04       184

    accuracy                           0.39      1409
   macro avg       0.21      0.21      0.18      1409
weighted avg       0.29      0.39      0.30      1409
```